CAPÍTULO 10
JAVA ENTERPRISE EDITION

A plataforma Java EE foi criada para o desenvolvimento de aplicações corporativas (baseada em componentes). Consiste de uma série de especificações bem detalhadas que, quando implementadas, podem ser usadas por desenvolvedores para tirar proveito e reutilizar infraestruturas prontas tais como: persistência em banco de dados, transação, acesso remoto, web services, gerenciamento de threads, gerenciamento de conexões HTTP, cache de objetos, gerenciamento da sessão web, balanceamento de carga, entre outros.

A ideia é que você possa criar uma aplicação que utilize esses serviços. Como esses serviços são bem complicados, você não perderá tempo implementando essa parte do sistema. Existem implementações tanto open source quanto pagas, ambas de boa qualidade.

10.1. SERVIDOR DE APLICAÇÃO

Tem a função de servir sua aplicação para auxiliá-la com serviços de infraestrutura. A própria Sun/Oracle desenvolve uma dessas implementações, o **Glassfish** que é open source e gratuito, porém não é o líder de mercado apesar de ganhar força nos últimos tempos.

Existem diversos servidores de aplicação famosos compatíveis com a especificação do J2EE 1.4, Java EE 5 e alguns do Java EE 6. O JBoss é um dos líderes do mercado e tem a vantagem de ser gratuito e open source. Alguns softwares implementam apenas uma parte dessas especificações do Java EE, como o Apache Tomcat, que só implementa JSP e Servlets, portanto não é totalmente correto chamá-lo de servidor de aplicação. Alguns dos servidores de aplicação mais conhecidos do mercado são:

Desenvolvedor	Implementação	Compatível
RedHat	JBoss Application Server	Java EE 5
Sun	GlassFish	Java EE 6
Apache	Apache Geronimo	Java EE 5
Oracle/BEA	WebLogic Application Server	Java EE 5
IBM	IBM Websphere Application Server	Java EE 5
Sun	Sun Java System Application Server	Java EE 5
SAP	SAP Application Server	Java EE 5

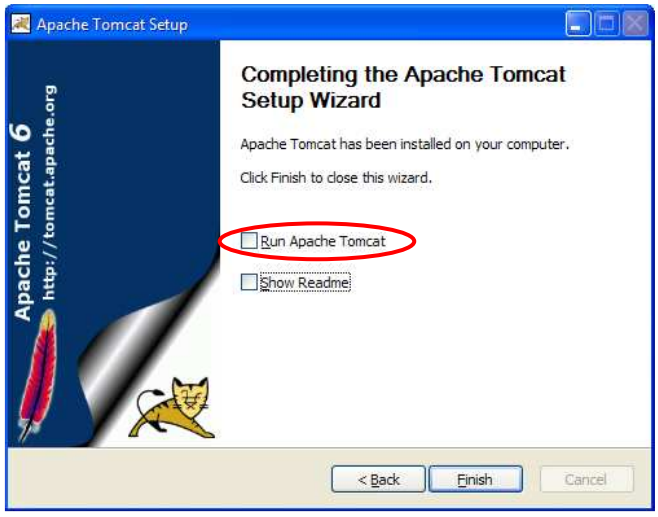
Um Servlet Container é um servidor que suporta essas funcionalidades, mas não necessariamente o Java EE completo, sendo indicado nos casos de não precisarmos de tudo do Java EE, apenas na parte web (a maioria das aplicações de médio porte se encaixa nessa

categoria). O mais famoso servlet containers é o Apache Tomcat, mas há outros como o Jetty e o Google App Engine.

10.2. INSTALANDO O APACHE TOMCAT

1) Baixe o Tomcat em http://tomcat.apache.org na página de downloads da versão que escolher.

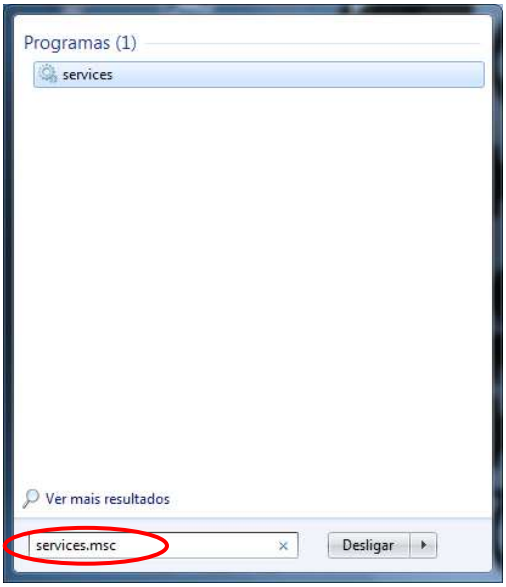
2) Para instalar o Tomcat no Windows basta executar o arquivo baixado. Antes de finalizar, desmarque a opção `Run Apache Tomcat`.



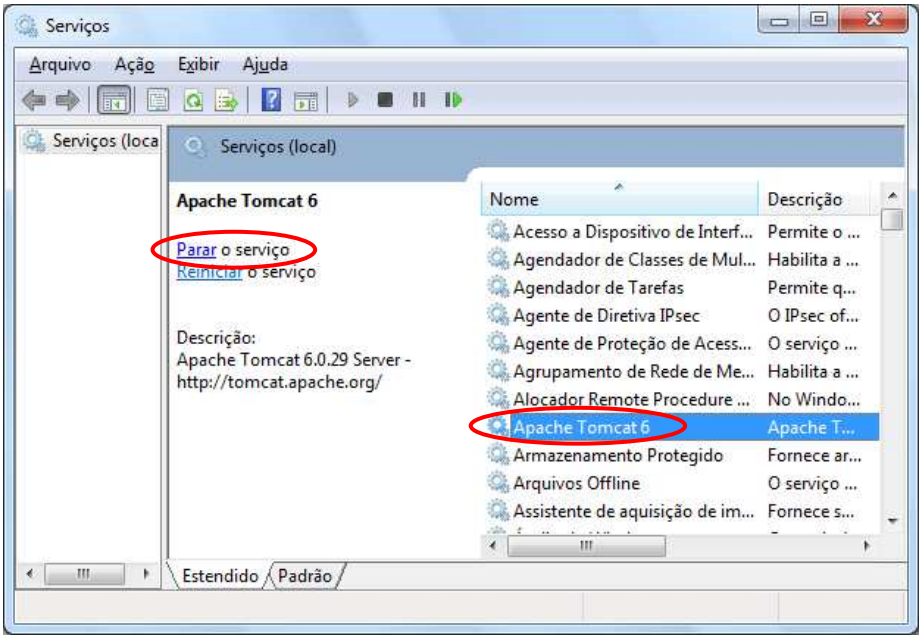
10.3. PARANDO SERVIÇO DO TOMCAT

Para executar o Tomcat no Eclipse é necessário, dependendo da instalação, **parar** o serviço para não ocasionar erro referente às portas de comunicação do sistema. Para tanto, acesse:

- Menu **Iniciar → Executar** e digite `services.msc`:



- Selecione a opção **Apache Tomcat** e clique na opção **Parar o serviço**. Com isso, o Tomcat será executado de dentro do Eclipse.

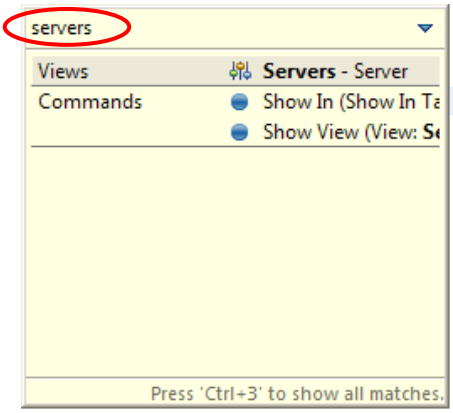


10.4. EXERCÍCIOS: Configurando o Tomcat no Eclipse

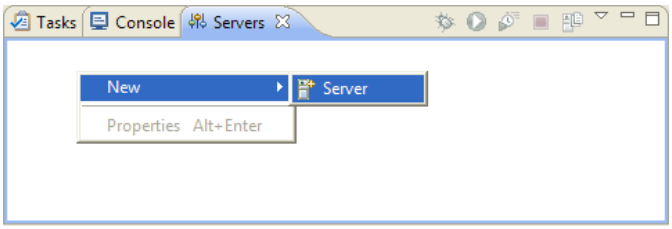
1) Pare o serviço do Tomcat.
   a. Acesse o menu **Iniciar → Executar**
   b. Digite `services.msc` e pressione **ENTER**
   c. Selecione **Apache Tomcat** e clique em **Parar o serviço**.

2) Acesse o Eclipse na Área de Trabalho.

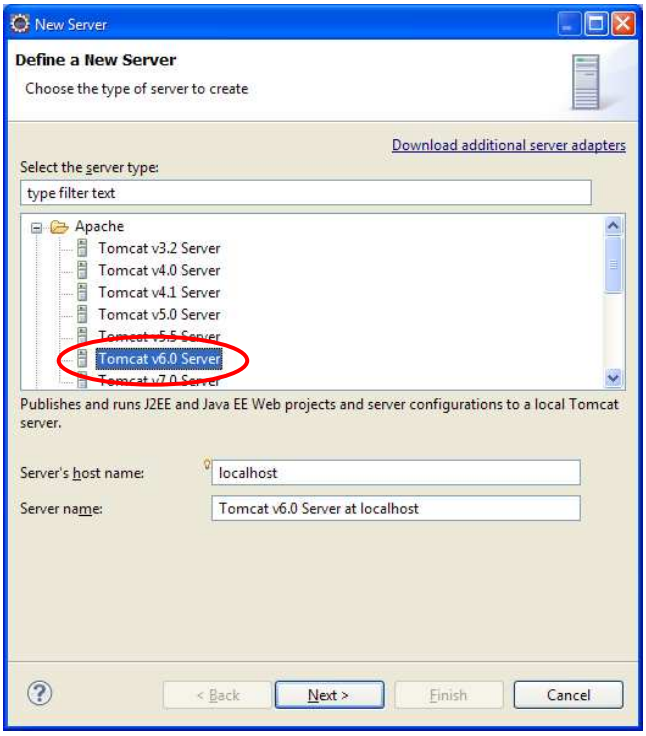
3) Abra a *View* de **Servers** no Eclipse. Aperte **Ctrl + 3** e digite `Servers`:



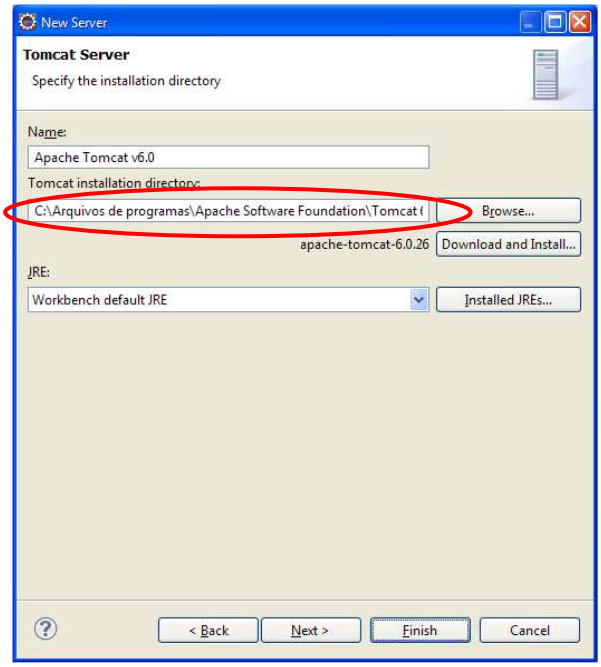
4) Clique com o botão direito dentro da aba Servers e vá em **New → Server**:



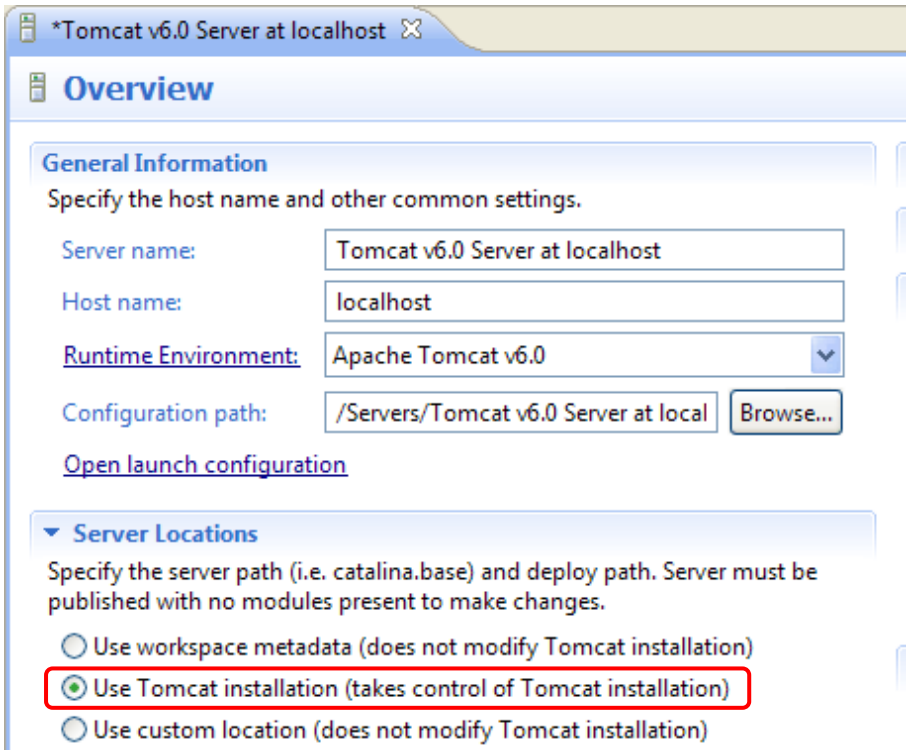
5) Selecione o **Apache Tomcat 6.0** e clique em **Next**:



6) Na próxima tela, selecione o diretório `C:\Arquivos de Programas\Apache Software Foundation\Tomcat` e clique em **Finish**.

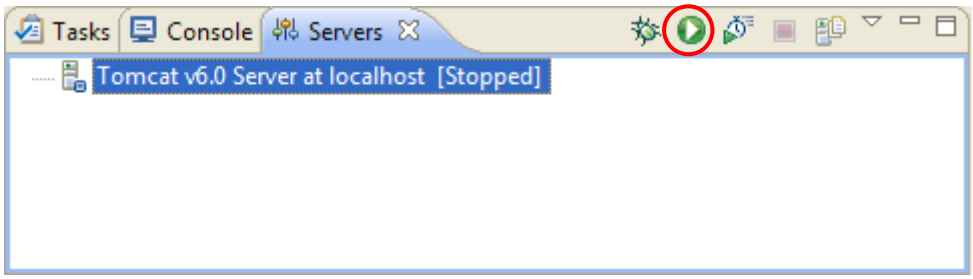


7) Por padrão, o Eclipse gerencia todo o Tomcat para nós e não permite que configurações sejam feitas fora da IDE. Para simplificar, vamos desabilitar essa configuração e deixar o Tomcat no modo padrão. Na aba Servers, dê dois cliques no servidor Tomcat que uma tela de configuração se abrirá. Localize a seção **Server Locations** e marque a opção **Use Tomcat installation**:

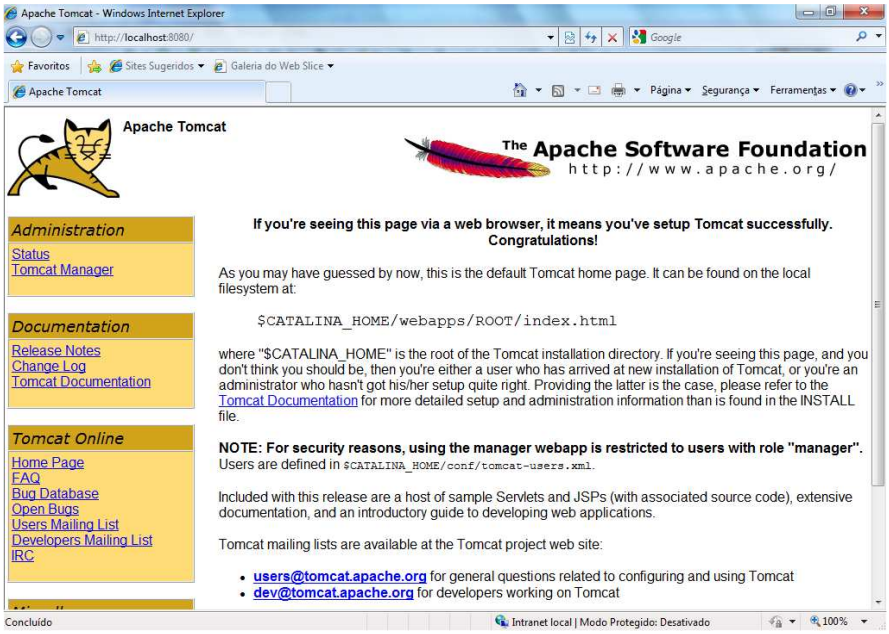


8) Salve e feche a tela.

9) Selecione o servidor que acabamos de adicionar e clique em **Start** (ícone play verde na view *servers*):

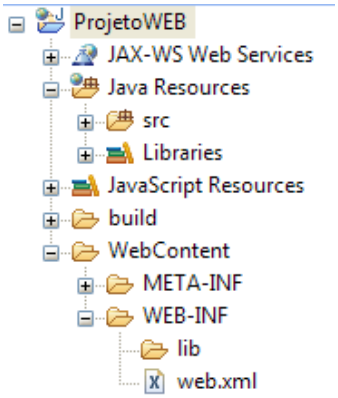


10) Abra o navegador e acesse a URL http://localhost:8080/. Deve aparecer uma tela de mensagem do Tomcat.



10.5. ESTRUTURA DE APLICAÇÕES WEB

Para sua aplicação ser reconhecida como aplicação WEB válida, esta deve se adequar a uma estrutura específica de diretório conforme a figura a seguir.



- **src**: é o local onde colocamos o nosso código fonte Java;
- **WebContent**: é a pasta raiz do conteúdo a ser exibido no navegador. Tudo que colocarmos na pasta WebContent será acessível na URL do projeto;
- **WEB-INF:** contém configurações e recursos necessários para o projeto rodar no servidor. Todo conteúdo da pasta WEB-INF não será visto a partir do navegador;
- **lib:** contém todas as bibliotecas necessárias para a aplicação;
- **web.xml:** armazena as configurações relativas a sua aplicação;
- **META-INF:** pasta opcional, mas gerada pelo Eclipse. É onde fica o arquivo de manifesto como usado em arquivos `.jar`.

Referência Bibliográfica:

Apostila Caelum - http://www.caelum.com.br/
http://www.globalcode.com.br/home
http://www.k19.com.br/downloads/apostilas-java
http://www.algaworks.com/
Barnes, David J. e Kolling, Michael. Programação Orientada a Objetos com Java. Pearson
Arnold, Ken, Gosling, James e Holmes, David. A Linguagem de Programação Java. Bookman
Harvey M. Deitel, Paul J. Deitel. Java Como Programar. 6ed. Pearson
Flanagan, David. Guia Essencial do Java. Bookman
Martin, Fowler. UML Essencial.
Livro de Informática – Centro Paula Souza